university of groningen

University Medical Center Groningen

University of Groningen

Complex-valued embeddings of generic proximity data

Münch, Maximilian; Straat, Michiel; Biehl, Michael; Schleif, Frank-Michael

Published in:
ArXiv

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Early version, also known as pre-print

Publication date:
2020

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Münch, M., Straat, M., Biehl, M., & Schleif, F-M. (2020). Complex-valued embeddings of generic proximity data. *ArXiv*.

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 26-12-2020

arXiv:2008.13454v1 [cs.LG] 31 Aug 2020

Complex-valued embeddings of generic proximity data*

Maximilian Münch[1,2][0000−0002−2238−7870], Michiel Straat[2][0000−0002−3832−978X], and Michael Biehl[2][0000−0001−5148−4568] Frank-Michael Schleif[1][0000−0002−7539−1283]

[1] University of Applied Sciences Würzburg-Schweinfurt, Department of Computer Science and Business Information Systems, D-97074 Würzburg, Germany {maximilian.muench, frank-michael.schleif}@fhws.de

[2] University of Groningen, Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, P.O. Box 407, NL-9700 AK Groningen, The Netherlands {m.biehl,m.j.c.straat@rug.nl}@rug.nl

Abstract. Proximities are at the heart of almost all machine learning methods. If the input data are given as numerical vectors of equal lengths, euclidean distance, or a Hilbertian inner product is frequently used in modeling algorithms. In a more generic view, objects are compared by a (symmetric) similarity or dissimilarity measure, which may not obey particular mathematical properties. This renders many machine learning methods invalid, leading to convergence problems and the loss of guarantees, like generalization bounds. In many cases, the preferred dissimilarity measure is not metric, like the earth mover distance, or the similarity measure may not be a simple inner product in a Hilbert space but in its generalization a Krein space. If the input data are non-vectorial, like text sequences, proximity-based learning is used or ngram embedding techniques can be applied. Standard embeddings lead to the desired fixed-length vector encoding, but are costly and have substantial limitations in preserving the original data's full information. As an information preserving alternative, we propose a complex-valued vector embedding of proximity data. This allows suitable machine learning algorithms to use these fixed-length, complex-valued vectors for further processing. The complex-valued data can serve as an input to complex-valued machine learning algorithms. In particular, we address supervised learning and use extensions of prototype-based learning. The proposed approach is evaluated on a variety of standard benchmarks and shows strong performance compared to traditional techniques in processing non-metric or non-psd proximity data.

Keywords: Proximity learning · embedding · complex values · learning vector quantizer.

* MM is supported by the ESF program WiT-HuB 4/2014-2020, project KI-trifft-KMU, StMBW-W-IX.4-6-190065. M.B. and M.S. acknowledge support through the Northern Netherlands Region of Smart Factories (RoSF) consortium, lead by Noordelijke Ontwikkelings en Investerings Maatschappij (NOM), The Netherlands, see http://www.rosf.nl

1 Introduction

Machine learning has a growing impact in various fields and the considered input data become more and more generic [27,15]. In particular non-vectorial data like text data, biological sequence data, graphs and other input formats are used [17]. The vast majority of learning algorithms are expecting fixed-length real value vector data as inputs and can not directly be used on non-standard data [27,28].

Using embedding approaches is one strategy to obtain a vectorial embedding, but this is costly and information is only partially preserved [27]. More recent approaches of deep learning like word2vec or others require the training of the embedding model and are only effective if the set of input data is large [12].

In a more generic scenario, proximity measures, like alignment functions, can be applied to compare non-vectorial objects to obtain a proximity score between two objects. These values can be used in the modeling step.

If all input data are pairwise compared, we obtain a proximity matrix $P \in \mathbf{R}^{N \times N}$. If the measure is a metric dissimilarity measure, we have a distance matrix that can be used for the nearest-mean classifier. In case of inner products like the euclidean inner product or the RBF-kernel function, a kernel matrix is obtained. If this kernel matrix is positive semidefinite (psd), a large number of kernel methods can be used in the modeling stage [35].

Also so-called empirical feature space approaches have been considered, but with the drawback of high model complexity and inherent data transformations [21]. For a more in-depth introduction into indefinite learning see [33,5]

Here we consider non-vectorial input data given either by a non-metric dissimilarity measure or a non-standard inner product, leading to an indefinite kernel function. As detailed in [33], learning models can be calculated on these generic proximity data in very different ways. Most often, the proximities are transformed to fit into classical machine learning algorithms, with a number of limitation [33]. In this work, we propose the application of a complex-valued embedding on these data. Recently different classical learning algorithms have been extended to complex-valued inputs [9]. It is now possible to preserve the information provided in the generic proximity data while learning in a fixed-length vector space using a highly effective, well-understood learning algorithm. The respective procedures are detailed in the following and evaluated on classical benchmark data with strong results.

2 Background and basic notation

Consider a collection of N objects $\mathbf{x}_i$, $i = \{1, 2, ..., N\}$, in some input space $\mathcal{X}$. Given a similarity function or inner product on $\mathcal{X}$, corresponding to a metric, one can construct a proper Mercer kernel acting on pairs of points from $\mathcal{X}$. For example, if $\mathcal{X}$ is a finite-dimensional vector space, a classical similarity function in this space is the Euclidean inner product (corresponding to the Euclidean distance).

2.1 Positive definite kernels - Hilbert space

The Euclidean inner product is also known as linear kernel with $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, where ϕ is the identity mapping. Another prominent kernel function is the RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right)$, with $\sigma > 0$ as a free scale parameter. In any case, it is always assumed that the kernel function $k(\mathbf{x}, \mathbf{x}')$ is psd.

The transformation ϕ is, in general, a *non-linear* mapping to a high-dimensional Hilbert space $\mathcal{H}$ and may not be given in an explicit form, but allowing *linear* techniques in $\mathcal{H}$. Instead of providing an explicit mapping, a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is given, which encodes the inner product in $\mathcal{H}$. The kernel k is a positive (semi-) definite function such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$, for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. The matrix $K_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ is an $N \times N$ kernel (Gram) matrix derived from the training data. For more general similarity measures, subsequently, we also use $\mathbf{S}$ to describe a similarity matrix.

Kernelized methods process the embedded data points in a feature space utilizing only the inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ [35], without the need to explicitly calculate ϕ, known as *kernel trick*. However, to employ the benefits of linear methods also explicit mappings of psd kernel function, using random Fourier features, are frequently used [30].

However, this assumption is not always fulfilled and the underlying similarity measure may not be metric and hence not lead to a Mercer kernel. Examples can be easily found in domain-specific similarity measures, as mentioned before. Such similarity measures imply *indefinite* kernels, preventing standard "kernel-trick" methods developed for Mercer kernels to be applied.

2.2 Non-positive definite kernels - Krein space

A Krein space is an *indefinite* inner product space endowed with a Hilbertian topology. Let $\mathcal{K}$ be a real vector space. An inner product space with an indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ on $\mathcal{K}$ is a bi-linear form where all $f, g, h \in \mathcal{K}$ and $\alpha \in \mathbb{R}$ obey the following conditions:

- Symmetry: $\langle f, g \rangle_{\mathcal{K}} = \langle g, f \rangle_{\mathcal{K}}$;
- linearity: $\langle \alpha f + g, h \rangle_{\mathcal{K}} = \alpha \langle f, h \rangle_{\mathcal{K}} + \langle g, h \rangle_{\mathcal{K}}$;
- $\langle f, g \rangle_{\mathcal{K}} = 0$ implies $f = 0$.

An inner product is positive semidefinite if $\forall f \in \mathcal{K}$, $\langle f, f \rangle_{\mathcal{K}} \geq 0$, negative definite if $\forall f \in \mathcal{K}$, $\langle f, f \rangle_{\mathcal{K}} < 0$, otherwise it is indefinite. A vector space $\mathcal{K}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ is called an inner product space.

An inner product space $(\mathcal{K}, \langle \cdot, \cdot \rangle_{\mathcal{K}})$ is a Krein space if we have two Hilbert spaces $\mathcal{H}_+$ and $\mathcal{H}_-$ spanning $\mathcal{K}$ such that $\forall f \in \mathcal{K}$ we have $f = f_+ + f_-$ with $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$ and $\forall f, g \in \mathcal{K}$, $\langle f, g \rangle_{\mathcal{K}} = \langle f_+, g_+ \rangle_{\mathcal{H}_+} - \langle f_-, g_- \rangle_{\mathcal{H}_-}$.

Indefinite kernels are typically observed by means of domain-specific non-metric similarity functions (such as alignment functions used in biology [36]), by specific kernel functions - e.g., the Manhattan kernel $k(x, x') = -||x - x'||_1$,

tangent distance kernel [14] or divergence measures, plugged into standard kernel functions [6]. A finite-dimensional Krein-space is a so-called pseudo-Euclidean space.

3 Embedding for non-psd proximities

Embedding of a proximity matrix into a vector space is not a new consideration, see e.g. [19], but was possible so far only in case of psd kernel functions. Given a symmetric *dissimilarity* matrix with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space, determined by the eigenvector decomposition of the associated similarity matrix $\mathbf{S}$, is always possible [11] [3]. Given the eigendecomposition of $\mathbf{S}$, $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, we can compute the corresponding vectorial representation $\mathbf{V}$ in the pseudo-Euclidean space by

$$\mathbf{V} = \mathbf{U}_{p+q+z} \left|\mathbf{\Lambda}_{p+q+z}\right|^{1/2} \tag{1}$$

where $\mathbf{\Lambda}_{p+q+z}$ consists of p positive, q negative non-zero eigenvalues and z zero eigenvalues. $\mathbf{U}_{p+q+z}$ consists of the corresponding eigenvectors. The triplet (p, q, z) is also referred to as the signature of the pseudo-Euclidean space. The crucial point in Eq. (1) is the *absolute* operator used in the embedding, which is also called a flip operation in the field of indefinite learning [33]. This effectively makes the representation of the data metric again and can have a substantial (potentially negative) impact on the data representation and modeling performance [23].

The transformation of dissimilarities to obey metric properties, or of similarities to be psd is in general expected to be useful. At least, it is technically useful because it permits to employ many mathematical concepts, as shown in [19], not available otherwise. We will go a step further and remove the absolute function from the embedding in Eq. (1) and obtain Eq. (2). Accordingly, the new embedding does not modify the data, in particular, an inner product of the embedded data reveals the input again.

$$\mathbf{V} = \mathbf{U}_{p+q+z}\mathbf{\Lambda}_{p+q+z}^{1/2} \tag{2}$$

Subsequently, we show how a number of mathematical operations can still be used to get an efficient machine learning model. This comes with additional effort and (finally) an embedding of the data into a *complex-valued* vector space is obtained. The embedding in Eq. (1) and Eq. (2) is straight forward but extremely costly.

Already in [19], this was addressed for psd kernels (only) by using the Nyström approximation. The approach in [19] can not be used directly in our setting since the input is non-psd.

[3] The associated similarity matrix can be obtained by double centering [27] of the dissimilarity matrix. $\mathbf{S} = -\mathbf{J}\mathbf{D}\mathbf{J}/2$ with $\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$, identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$.

In our former work [10], we have shown that the Nyström approximation remains valid for generic proximity data, in particular similarities that are non-psd. This has been recently reconsidered by a simplified proof in [26]. Hence the Nyström approximation becomes available also to approximate a non-psd matrix. In our work [10], we have further shown how the Nyström approximation can also be used to have an approximated Double-Centering to deal with dissimilarity data. Our former work helps in two ways to permit the embedding of Eq. (2) effectively:

1. the input needs not to be a kernel but can also be a dissimilarity matrix
2. the Nyström matrix approximation can also be done for non-psd similarities which reduced the costs of the embedding

In the Nyström approximation, we have to specify the number of m landmarks with $m \ll N$. The landmarks can be selected for non-psd matrices randomly or by kmeans++ as shown recently in [25,26]. Our efficient approach to get an approximated complex-valued, vectorial embedding of a non-psd matrix is shown in Algorithm 1.

Algorithm 1 Approximated embedding of symmetric proximities

Embed_proximities(P, m)
if P is dissimilarity **then**
  $Knm, Kmm :=$ ApproximatedDoubleCentering(P, m) using [10] and kmeans++
else
  $Knm, Kmm :=$ Approximate(P, m) using [10] for similarities and kmeans++
end if
$[C, A] :=$ eig(Kmm); with eigenvectors C and eigenvalues in A (diagonal)
$W :=$ diag(sqrt(1./diag(A)))*C' modified Nyström projection matrix
$M :=$ W*Knm' complex-valued embedding
$K^* :=$ M'*M reconstruction
return M

In the first step of Algorithm 1, the input matrix is approximated using the Nyström approximation (potentially with an integrated double centering). This can be done with linear costs and with guaranteed approximation bounds [10,26]. Subsequently, we calculate the essential part of the embedding function in Eq. (2) combined with the projection matrix of the Nyström approximation, by taking the square root of the (pseudo-) inverse of the eigenvalue decomposition of Kmm. This can be done with linear costs as shown in [10]. The vectorial embedding M is finally done by mapping the rectangular Nyström part Knm of the similarities to the projection matrix W[4]. The embedding is complex-valued if the similarity matrix K is non-psd and hence A contains negative values.

[4] The step how the projection matrix $W*$ is calculated, is slightly related to so-called Landmark MDS as suggested in [2] for psd-only matrices.

We now have an approximated complex-valued fixed-length vectorial embedding of the proximity data P whereby the respective reconstruction is exact if the rank of P equals to the number of non-vanishing eigenvalues in A. Algorithm 1 has a linear complexity as long as the number of landmarks $m \ll N$, which is in general the case. In contrast to many other methods [32], the embedding procedure has a straight forward out of sample extension. The mapping in 1 can be done for new points by evaluating the proximity function for the landmark point and using the respective projection function.

For the complex-valued embedding (so far) a quite limited number of machine learning algorithm is available, like the complex-valued support vector machine (cSVM) [41,3], the complex-valued generalized learning vector quantization (cGMLVQ) [4,37], or a complex-valued neural network (cNN) [13,38,7]. Further, a nearest neighbor (NN) classifier by employing a standard norm operator can be used. While cSVM, cGMLVQ, cNN are parametric methods, the NN classifier is parameter-free and can be used directly. In particular, after applying the norm, the obtained dissimilarity values are metric. For the cGMLVQ, we briefly recap the respective derivations and show how it can be effectively used within our setting.

4 Complex-valued Generalized Learning Vector Quantization

In this section, we will review the *complex-valued Learning Vector Quantization* (cGLVQ). We first lay out the general idea of LVQ classification and the learning rules in modern variants. We will then review the learning rules' adaptation to make the algorithm suitable for handling complex-valued data.

In Learning Vector Quantization (LVQ), the classification scheme is parameterized by a set of labeled prototypes and a distance measure $d(\cdot, \cdot)$. New data is classified according to the label of the nearest prototype with respect to the distance measure $d(\cdot, \cdot)$. This original idea was proposed by Kohonen in 1986. In contrast to the k-nearest neighbor classifier in which the full dataset is used in the classification procedure, the classes in LVQ schemes are represented by only very few prototypes. Hence, in the algorithm's working phase, LVQ takes less computational effort and storage in order to assign class labels to new data. Moreover, LVQ is often praised for its white-box character: The prototypes that are crucial for the classification represent typical examples of the classes in the dataspace. Numerous extensions of LVQ that have been proposed over the years, such as adaptive distance measures known as *relevance learning*, low-rank approaches a.s.o, further increasing the performance and interpretability of the method. The interpretability of LVQ provides important insights in many applications, see for instance [39,1] for examples in the medical domain.

4.1 Training an LVQ classifier

Given a training dataset of N labeled objects in K classes, i.e. $(\mathbf{x}_i, y_i)$ with $i = \{1, 2, ..., N\}$, in which $\mathbf{x}_i \in \mathcal{X}$ is an input object and $y_i \in \{1, 2, ..., K\}$ its class

label. The aim of the training procedure is choosing prototypes $\{(\mathbf{w}_k, y_k) \,|\, 1 \leq k \leq M\}$ in the space of the objects $\mathcal{X}$, such that the resulting classification scheme gives high classification accuracy with respect to unseen data. The number of prototypes per class is a hyperparameter chosen by the user and one has to choose at least one prototype per class, hence the total number of prototypes M is at least the number of classes K, i.e. $K \geq M$. The distance measure $d(\cdot, \cdot)$ is of central importance in the training- and classification procedure. It is used to compare input objects with prototype vectors for classification and for determining the change in positions of the prototypes in the training procedure. If the data space $\mathcal{X}$ is Euclidean, the squared Euclidean distance measure is a common choice:

$$d^{\Lambda}(\mathbf{w}, \mathbf{x}_i) = (\mathbf{x}_i - \mathbf{w})^H \mathbf{\Omega}^H \mathbf{\Omega} (\mathbf{x}_i - \mathbf{w}) \,, \tag{3}$$

in which $\mathbf{\Omega}$ is the identity matrix and H is the Hermitian transpose. One drawback of this choice is that it is affected by differing scales of the input variables x_i, such that an x_i with a large standard deviation compared to others has a large influence in the distance measure. This is, however, easily prevented by normalizing all input variables to unit standard deviation. A major improvement in LVQ is the introduction of *relevance learning* [16,34], in which the elements of the linear projection $\mathbf{\Omega}$ are adapted during training to reflect the importance of the features in the classification task and to account for correlations between features.

The original idea of Kohonen's LVQ1 is to update the prototypes based on a heuristic Winner Takes All (WTA) rule. At the random presentation of an example, its closest prototype is attracted by the example if the class labels coincide and is repelled from the example otherwise. In [31], the authors proved that this training scheme does not converge and instead the authors propose a by now popular and successful cost function that does converge in the gradient descent. The cost for an example is defined as:

$$E_{GLVQ} = \sum_{i=1}^{P} \Phi(\mu_i), \text{ with } \mu_i = \frac{d_+(\mathbf{x}_i) - d_-(\mathbf{x}_i)}{d_+(\mathbf{x}_i) + d_-(\mathbf{x}_i)} \,. \tag{4}$$

The argument μ_i is based on the difference between the distance $d_+(\mathbf{x}_i)$ from its position to the closest prototype with the same label and the distance $d_-(\mathbf{x}_i)$ to the closest prototype with a different label, normalized to the range $\mu_i \in [-1, 1]$. The function $\Phi(\cdot)$ is monotonically increasing and is usually chosen to be identity $\Phi(x) = x$ or the logistic function $\Phi(x) = 1/(1 + \exp(-kx))$. The above cost function can be minimized with respect to the prototypes $\mathbf{w}$ and the relevance matrix $\mathbf{\Omega}$ by either batch- or stochastic gradient descent. To formulate the update rule with respect to $\mathbf{w}_\pm$ and $\mathbf{\Omega}$ for the example $\mathbf{x}_i$, one applies the chain rule:

$$\mathbf{w}_\pm = \mathbf{w}_\pm - \alpha \Phi'(\mu_i) \frac{\partial \mu_i}{\partial d_\pm} \frac{\partial d_\pm}{\partial \mathbf{w}_\pm} \,, \quad \mathbf{\Omega} = \mathbf{\Omega} - \beta \Phi'(\mu_i) \frac{\partial \mu_i}{\partial d} \frac{\partial d}{\partial \mathbf{\Omega}} \tag{5}$$

In case $\Phi(\cdot)$ is a zero-centered sigmoidal function, the largest updates occur for examples close to the decision boundary with $\mu_i \approx 0$, which causes the classifier to learn faster from the most informative examples.

4.2 Learning rules for complex-valued data

In the complex-valued data space $\mathbb{C}^N$, the squared distance in Eq. (3) between an object $\mathbf{x}_i$ and an example $\mathbf{w}_\pm$ is always real: It is the sum of the squared magnitudes of the components of the projected difference vector $\mathbf{\Omega}(\mathbf{x}_i - \mathbf{w}_\pm)$. Therefore only the innermost derivatives of the distance measure in Eq. (5) are with respect to the complex-valued variables. These can be done in an elegant way using the Wirtinger differential operators [40] as proposed in [9]:

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \quad \frac{\partial}{\partial z^*} = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right), \tag{6}$$

in which $z = x+iy$ and $z^* = x-iy$, the complex conjugate. Using the differential operator with respect to the conjugate of the complex variable, the inner most derivatives in Eq. (5) are as follows:

$$\frac{\partial d*}{\partial \mathbf{w}_\pm^*} = -\mathbf{\Omega}^H\mathbf{\Omega}(\mathbf{x}_i - \mathbf{w}_\pm), \quad \frac{\partial d}{\partial \mathbf{\Omega}^*} = \mathbf{\Omega}(\mathbf{x}_i - \mathbf{w}_\pm)(\mathbf{x}_i - \mathbf{w}_\pm)^H, \tag{7}$$

which are conceptually very similar to the derivatives for real-valued variables.

5 Experiments

In this section, we show the effectiveness of the proposed embedding approach on a set of benchmark data typically used in the area of proximity-based supervised learning. The following section contains a brief description of the datasets with details in the references. Subsequently, we evaluate the performance of our embedding approach on these datasets compared to some baseline classifier.

5.1 Datasets

We use multiple standard benchmark data for similarity-based learning. All data sets used in this experimental setup are indefinite with different spectral properties. If the data are given as dissimilarities, a corresponding similarity matrix can be obtained by double centering [27]: $S = -JDJ/2$ with $J = (I - \mathbf{1}\mathbf{1}^\top/N)$, with identity matrix I and vector of ones $\mathbf{1}$. The datasets used for the experiments are described in the following and summarized in Table 1, with details given in the references. The triplet (p, q, z) is also referred to as the signature. In this context, the signature describes the ratio of positive to negative and zero eigenvalues of the respective data set.

1. **Balls3d/Balls50d** consist of 200/2000 samples in two/four classes. The dissimilarities are generated between two constructed balls using the shortest distance on the surfaces. The original data description is provided in [29].
2. The Copenhagen **Chromosomes** data set constitutes 4,200 human chromosomes from 21 classes represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance. Details are provided in [24].

Dataset	#samples	#classes	signature
Balls3d	200	2	$(48, 152, 0)$
Balls50d	2000	2	$(853, 1147, 0)$
Chromosomes	$4,200$	21	$(2258, 1899, 43)$
DelftGestures	$1,500$	20	$(963, 536, 1)$
Protein	213	4	$(170, 40, 3)$
Sonatas	1068	5	$(1063, 4, 1)$
Zongker	2000	10	$(1039, 961, 0)$

Table 1: Overview of the datasets used in our experimental setup. Details are given in the textual description.

3. The **Delft gestures** (1500 points, 20 classes, balanced, signature: (963,536,1)), taken from [8], is a set of dissimilarities generated from a sign-language interpretation problem (see Figures 8 to 10c). It consists of 1500 points with 20 classes and 75 points per class. The gestures are measured by two video cameras observing the positions of the two hands in 75 repetitions of creating 20 different signs. The dissimilarities are computed using a dynamic time-warping procedure on the sequence of positions (Lichtenauer, Hendriks, Reinders, 2008).
4. **Protein**: the Protein data set has sequence-alignment similarities for 213 proteins and is used for comparing and classifying protein sequences according to its four classes of globins: heterogeneous globin (G), hemoglobin-A (HA), hemoglobin-B (HB) and myoglobin (M). The signature is (170,40,3), where class one through four contains 72, 72, 39, and 30 points, respectively [18].
5. **Sonatas** dataset consists of 1068 sonatas from five composers (classes) from two consecutive eras of western classical music. The musical pieces were taken from the online MIDI database *Kunst der Fuge* and transformed to similarities by normalized compression distance [22].
6. **Zongker** dataset is a digit dissimilarity dataset. The dissimilarity measure was computed between 2000 handwritten digits in 10 classes, with 200 entries in each class [20].

5.2 Results

In this section, we evaluate the performance of the proposed embedding on the mentioned datasets using a model that is able to handle complex-valued data. For this purpose, we use the Generalized Learning Vector Quantization (GLVQ) from Sec. 4 with the learning rule for complex-valued data from Sec. 4.2. The GLVQ was parametrized once with and once without relevance learning. Within the GLVQ, we verified that the corrected input matrix was indeed psd by an additional test using an eigendecomposition, no fails were found. For the initialization of prototypes, we used one prototype per class for the G(M)LVQ. In the embedding step of Algorithm 1 we set the meta parameter m (# of landmarks) by a rule of thumb:

- If the number of data points in the data set is < 1000, our recommendation is $m = 40$.
- For data sets with a size $1000 < N < 5000$, we recommend $m = 70$.
- otherwise $m = 100$.

Additionally, to the GLVQ, we chose a nearest neighbor classifier (*NN*) for performance comparisons to our proposed method, which remains valid also for generic psd data. In the nearest neighbor, the kernel matrix was left in the original - that means in the uncorrected, indefinite - state.

Experiments were run in a ten-fold cross-validation. Mean prediction accuracy on the hold out test data and the respective standard deviation is reported and shown in Table 2.

Dataset	CGLVQ	CGMLVQ	Nearest Neighbor
Balls3d	0.61 ± 0.07	$\mathbf{0.99 \pm 0.03}$	0.48 ± 0.07
Balls50d	0.29 ± 0.04	$\mathbf{0.50 \pm 0.06}$	0.25 ± 0.02
Chromosomes	0.92 ± 0.01	0.94 ± 0.01	$\mathbf{0.95 \pm 0.02}$
DelftGestures	0.94 ± 0.02	$\mathbf{0.96 \pm 0.02}$	0.95 ± 0.01
Protein	0.91 ± 0.07	$\mathbf{0.98 \pm 0.05}$	0.22 ± 0.04
Sonatas	0.82 ± 0.03	0.89 ± 0.03	$\mathbf{0.90 \pm 0.01}$
Zongker	0.87 ± 0.03	$\mathbf{0.92 \pm 0.02}$	0.58 ± 0.05

Table 2: Prediction accuracy (mean ± standard-deviation) for the cGLVQ variants and the nearest neighbor classifier. Column *cGLVQ* shows the performance of the Generalized Learning Vector Quantization without relevance learning and column *cGMLVQ* provides the performance of the complex-valued Generalized Matrix Learning Vector Quantization, employing relevance learning. Column *Nearest Neighbor* shows the performance of the baseline classification.

If the data were left uncorrected we obtained often a rather poor result using the nearest neighbor classifier, sometimes even significantly worse compared to cGLVQ and cGMLVQ (see balls3d, balls50d, protein, zongker). In some cases, NN had equal or slightly better performance than the two cG(M)LVQ variants (Chromosomes, Sonatas). This is due to the spectrum of eigenvalues: Chromosomes has many eigenvalues, which are almost negligible and close to zero. Sonatas has only a few negative eigenvalues and these eigenvalues are also close to zero. With respect to the two cGLVQ variants, one can also see a difference in performance between GLVQ and GMLVQ: the activation of relevance learning within the cGMLVQ leads to significantly better results in some cases. However, even the mere use of the cGLVQ without relevance learning leads to a significant increase compared to the NN with uncorrected data. Therefore, we assume that a correction step, like our embedding approach, is indeed essential since the use of uncorrected non-psd data shows a clear drop in accuracy.

In summary, the presented approach, applying an embedding of the indefinite input data into a complex-valued vector space, shows promising results on a variety of data sets.

6 Conclusions

In this work, we proposed a complex-valued embedding and processing pipeline to analyze non-metric or non-psd proximity data. The approach shows very promising performance on a variety of datasets and is easy to employ. A careful combination of approximation techniques, derived by the authors in former work, permits a valid and still effective calculation of the embedding matrix. By processing the embedding matrix, a straight forward out of sample extension is obtained, which is not easily available for many traditional indefinite learning approaches. The low-rank embedding has the benefit that the reconstructed matrix approximates the original indefinite kernel with low error, hence all major information in the original data is preserved. In the classifier models, a norm operator is used which eventually leads to a psd dissimilarity representation. The effect of this operation, which is not equivalent to a classical flip correction on the original indefinite kernel, has to be evaluated in more detail in future work. Using learning algorithms for complex-valued embeddings, predictive models can be obtained with low computational costs. In this initial work, we focused on complex-valued G(M)LVQ and Nearest Neighbor to calculate classification model, but this will be extended to further modeling approaches in future work. Our initial findings show that the suggested complex-valued embedding of indefinite proximity data, combined with complex-valued classifier models is a promising very effective approach to more complicated alternatives.

References

1. Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors. Journal of Clinical Endocrinology and Metabolism **96**(12), 3775–3784 (12 2011). https://doi.org/10.1210/jc.2011-1565
2. Belongie, S.J., Fowlkes, C.C., Chung, F.R.K., Malik, J.: Spectral partitioning with indefinite kernels using the nyström extension. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part III. Lecture Notes in Computer Science, vol. 2352, pp. 531–542. Springer (2002). https://doi.org/10.1007/3-540-47977-5_35, https://doi.org/10.1007/3-540-47977-5_35
3. Bouboulis, P., Theodoridis, S., Mavroforakis, C., Evaggelatou-Dalla, L.: Complex support vector machines for regression and quaternary classification. IEEE Transactions on Neural Networks and Learning Systems **26**(6), 1260–1274 (2015)
4. Bunte, K., Schleif, F.M., Biehl, M.: Adaptive learning for complex-valued data. In: Proceedings of ESANN 2012. pp. 387–392 (2012)
5. Chen, L., Lian, X.: Efficient similarity search in nonmetric spaces with local constant embedding. IEEE Trans. Knowl. Data Eng. **20**(3), 321–336 (2008)

6. Cichocki, A., Amari, S.I.: Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. Entropy **12**(6), 1532–1568 (2010)
7. Dramsch, J.S., Lüthje, M., Christensen, A.N.: Complex-valued neural networks for machine learning on non-stationary physical data. CoRR **abs/1905.12321** (2019), http://arxiv.org/abs/1905.12321
8. Duin, R.P.: PRTools (march 2012), http://www.prtools.org
9. Gay, M., Kaden, M., Biehl, M., Lampe, A., Villmann, T.: Complex variants of glvq based on wirtinger's calculus. In: Merényi, E., Mendenhall, M.J., O'Driscoll, P. (eds.) Advances in Self-Organizing Maps and Learning Vector Quantization. pp. 293–303. Springer International Publishing, Cham (2016)
10. Gisbrecht, A., Schleif, F.: Metric and non-metric proximity transformations at linear costs. Neurocomputing **167**, 643–657 (2015)
11. Goldfarb, L.: A unified approach to pattern recognition. Pattern Recognition **17**(5), 575 – 582 (1984)
12. Goodfellow, I.J., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge, MA, USA (2016), http://www.deeplearningbook.org
13. Guberman, N.: On complex valued convolutional neural networks (2016)
14. Haasdonk, B., Keysers, D.: Tangent distance kernels for support vector machines. In: ICPR (2). pp. 864–868 (2002)
15. Halpin, H., McNeill, F.: Discovering meaning on the go in large heterogenous data. Artif. Intell. Rev. **40**(2), 107–126 (2013). https://doi.org/10.1007/s10462-012-9377-4, https://doi.org/10.1007/s10462-012-9377-4
16. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks **15**(8), 1059 – 1068 (2002). https://doi.org/https://doi.org/10.1016/S0893-6080(02)00079-5
17. Hendler, J.A.: Data integration for heterogenous datasets. Big Data **2**(4), 205–215 (2014). https://doi.org/10.1089/big.2014.0068, https://doi.org/10.1089/big.2014.0068
18. Hofmann, T., Buhmann, J.M.: Pairwise data clustering by deterministic annealing. IEEE Trans. Pattern Anal. Mach. Intell. **19**(1), 1–14 (1997)
19. Iosifidis, A., Gabbouj, M.: Nyström-based approximate kernel subspace learning. Pattern Recognition **57**, 190–197 (2016)
20. Jain, A., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(12), 1386–1391 (1997)
21. Kar, P., Jain, P.: Similarity-based learning via data driven embeddings. In: Proc. of Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Granada, Spain. pp. 1998–2006 (2011)
22. Mokbel, B.: Dissimilarity-based learning for complex data. Ph.D. thesis, Bielefeld University (2016), http://nbn-resolving.de/urn:nbn:de:hbz:361-29004254
23. Münch, M., Raab, C., Biehl, M., Schleif, F.: Structure preserving encoding of non-euclidean similarity data. In: Marsico, M.D., di Baja, G.S., Fred, A.L.N. (eds.) Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2020, Valletta, Malta, February 22-24, 2020. pp. 43–51. SCITEPRESS (2020)
24. Neuhaus, M., Bunke, H.: Edit distance based kernel functions for structural pattern classification. Pattern Recognition **39**(10), 1852–1863 (2006)

25. Oglic, D., Gärtner, T.: Nyström method with kernel k-means++ samples as landmarks. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 2652–2660. PMLR (2017), http://proceedings.mlr.press/v70/oglic17a.html
26. Oglic, D., Gärtner, T.: Scalable learning in reproducing kernel krein spaces. In: Proc. of the 36th Int. Conf. on ML, ICML 2019, Long Beach, California, USA. pp. 4912–4921 (2019)
27. Pekalska, E., Duin, R.: The dissimilarity representation for pattern recognition. World Scientific (2005)
28. Pekalska, E., Duin, R.P.W., Günter, S., Bunke, H.: On not making dissimilarities euclidean. In: SSPR&SPR 2004. pp. 1145–1154 (2004)
29. Pekalska, E., Harol, A., Duin, R.P.W., Spillmann, B., Bunke, H.: Non-euclidean or non-metric measures can be informative. In: Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2006 and SPR 2006, Hong Kong, China, August 17-19, 2006, Proceedings. pp. 871–880 (2006)
30. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: Platt, J.C., Koller, D., Singer, Y., Roweis, S.T. (eds.) Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007. pp. 1177–1184. Curran Associates, Inc. (2007), http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines
31. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Proceedings of the 8th International Conference on Neural Information Processing Systems. p. 423–429. NIPS'95, MIT Press, Cambridge, MA, USA (1995)
32. Schleif, F.M.: Generic probabilistic prototype based classification of vectorial and proximity data. Neurocomputing **154**, 208–216 (2015)
33. Schleif, F., Tiňo, P.: Indefinite proximity learning: A review. Neural Computation **27**(10), 2039–2096 (2015)
34. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization **21**(12), 3532–3561 (Dec 2009). https://doi.org/10.1162/neco.2009.11-08-908, https://doi.org/10.1162/neco.2009.11-08-908
35. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis and Discovery. Cambridge University Press (2004)
36. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. Journal of molecular biology **147**(1), 195–197 (Mar 1981)
37. Straat, M., Kaden, M., Gay, M., Villmann, T., Lampe, A., Seiffert, U., Biehl, M., Melchert, F.: Learning vector quantization and relevances in complex coefficient space. Neural Computing and Applications (Mar 2019)
38. Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y., Pal, C.J.: Deep complex networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=H1T2hmZAb
39. van Veen, R., Gurvits, V., Kogan, R.V., Meles, S.K., de Vries, G.J., Renken, R.J., Rodriguez-Oroz, M.C., Rodriguez-Rojas, R., Arnaldi, D., Raffa, S., de Jong, B.M., Leenders, K.L., Biehl, M.: An application of generalized matrix learning vector quantization in neuroimaging. Computer Methods and Programs in Biomedicine p. 105708 (2020). https://doi.org/https://doi.org/10.1016/j.cmpb.2020.105708, http://www.sciencedirect.com/science/article/pii/S0169260720315418

40. Wirtinger, W.: Zur formalen theorie der funktionen von mehr komplexen veränderlichen. Mathematische Annalen **97**, 357–376 (1927)
41. Zhang, L., Zhou, W., Jiao, L.: Complex-valued support vector classifiers. Digital Signal Processing **20**(3), 944 – 955 (2010)